REMARKS

Claims 33-36 have been added.

Claims 1-32 stand rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,581,205 to Cochrane et al. (hereafter Cochrane) in view of U.S. Patent 6,735,587 to Colby et al. (hereafter Colby). Applicants respectfully traverse.

1. "Claim 1 recites "[a] method for applying a row from a source table to a destination table, the method comprising: selecting a first column from a source table; selecting a second column from a destination table; performing an outer join operation on the source table and the destination table using the first and second columns; updating each row in the destination table with a row from the results of the outer join operation containing a matching element in the first and second columns; and inserting into the destination table each row from the results of the outer join operation with a non-matching element in the first and second columns, the method performing no more than one scan per table." Conversely, Cochrane teaches the results of a specification query are joined to the summary table, and the use of multiple scans of the tables via the separate join, and the update, delete, and insert operations. For example, Cochrane teaches:

> FIG. **4** is a QGM that illustrates the operation of the present invention. Summary tables **400** (or materialized views) are based on the results of a specification query **402** that condenses, combines, or otherwise processes one or more underlying base tables **404**. When the underlying base tables **404** are modified, the summary table **400** must also be maintained to accurately reflect the modified tables. The results of the specification query **402** are joined **406** to the summary table **400** to determine whether an UPDATE **408**, DELETE **410**, or INSERT **412** operation is required to maintain the summary table **400**.

(Cochrane, col. 5, lines 9-19 and FIG. 4) underline added.

The above passage discloses that "results of a *specification query* are *joined* with the summary table." However, the claim includes a source table not a specification query, and an outer join not a join. The passage further discloses that separate UPDATE, DELETE and INSERT operations are used which require additional scans of the tables, in contrast to the claim which states one scan per table. As such, Cochrane does not disclose, teach, or suggest "[a]

method for applying a row from a source table to a destination table, the method comprising: selecting a first column from a source table; selecting a second column from a destination table; performing an outer join operation on the source table and the destination table using the first and second columns; updating each row in the destination table with a row from the results of the outer join operation containing a matching element in the first and second columns; and inserting into the destination table each row from the results of the outer join operation with a non-matching element in the first and second columns, the method performing no more than one scan per table" as recited in claim 1.

Colby does not cure this deficiency. Colby teaches tagging, aggregating, and matching information in a pre-computed aggregate view of a materialized aggregate table using various working data sets and using that information to update (insert, remove, update) the materialized aggregate table. For example, Colby teaches:

> **FIG. 2** shows another method in accordance with the invention for incrementally maintaining a pre-computed aggregate view that is self-maintainable. The system receives the pre-computed aggregate view V that is self-maintainable, including its view definition V.sub.D and materialized aggregate table MV (step **202**). The system also receives changes to a base table of the pre-computed aggregate view, the changes being represented as deletions and insertions (step **204**). The system tags insertions and deletions with distinguishing literals and combines them to produce .delta.F (deltaF) (step **206**). The system computes aggregations on .delta.F to produce aggregated change set .delta.G (deltaG) (step **208**). The system matches rows from .delta.G with rows in MV and produces .delta.J (deltaJ), which contains all rows from .delta.G with matched rows tagged with information from corresponding rows in MV and further contains non-matching rows from .delta.G tagged as non-matching (step **210**). (Matched rows are rows that have corresponding rows and non-matching rows are rows that do not have corresponding rows.) One way to tag rows in .delta.J as non-matching is to insert NULL values in MV columns. The system produces .delta.J' (deltaJ') by selecting from .delta.J rows identified as either: (i) matched rows or (ii) identified as non-matching but resulting from more base table changes that are insertions than deletions for the aggregated group represented by the row (step **212**). The system inserts new rows into MV, the new rows being constructed from rows (in .delta.J') identified as non-matching rows (step **214**). The system removes the identified rows from .delta.J' to produce .delta.I (deltaI) (also step **214**). The system deletes from MV rows matching rows in .delta.I, each row representing an aggregated group that has as many base table deletions as the sum of the number of base table insertions and the number of pre-modified base table rows for that group as stored in MV (step **216**). The system removes these identified rows from .delta.I to produce .delta.D (deltaD) (also step 216). The system updates rows in MV matching rows

in .delta.D (step **218**).
(Colby, col. 7, line 62-col. 8, line 30) underline added.

This cites passage of Colby teaches aggregation, tagging, and matching of information from a materialized aggregated table and performing multiple operations on the MV table. In other words, Colby teaches manipulating the materialized view data into delta data sets and in turn performing *multiple operations* (inserts, removes, and updates) *on a materialized view* but does not disclose, teach, or suggest "[a] method for applying a row from a source table to a destination table, the method comprising: selecting a first column from a source table; selecting a second column from a destination table; performing an outer join operation on the source table and the destination table using the first and second columns; updating each row in the destination table with a row from the results of the outer join operation containing a matching element in the first and second columns; and inserting into the destination table each row from the results of the outer join operation with a non-matching element in the first and second columns, the method performing no more than one scan per table" as recited in claim 1. As neither Cochrane nor Colby, disclose, teach, or suggest all the limitations in claim 1, Cochrane, and Colby, can not be used be used to preclude patentability of claim 1 under 35 U.S.C. § 103.

2. Claim 17 recites sufficiently similar limitations as claim 1, and is therefore, patentable over Cochrane and Colby for at least the same reasons.

3. Claims 2-4 and 18-20 depend on claims 1 and 17, and are therefore, patentable over Cochrane and Colby for at least the same reasons.

4. Claim 5 recites "[a] statement to insert a new row or update an existing row in a database table, the statement implementing a process comprising the steps of: selecting from a source table a first column comprising a plurality of elements; selecting from a destination table a second column comprising a plurality of elements; determining a set of matching rows based upon the success of a comparison operation on an element in the first column and an element in the second column; determining a set of non-matching rows based upon the failure of a comparison operation on the first column element and the second column element; updating the destination table with the set of matching rows; and inserting into the destination table the set of

non-matching rows, the statement comprising a single query language statement." Conversely Cochrane teaches a multiple transaction method. For example, Cochrane teaches:

> Consider Scenario #1a as a replacement for Scenario #1. Transaction #1 reads the materialized view, discovers that the 1998 record does not exist, and obtains an U-lock on either the next-key (e.g., the 1999 record) or EOF if the next-key does not exist. Transaction #1 then computes the value for the 1998 record based on Transaction #1 transition values. Transaction #1 obtains an X-lock on the next-key or EOF in the materialized view, inserts the new record into the materialized view, commits.and releases its U- and X-locks. Transaction #2 reads the materialized view, sees the 1998 record, obtains an U-lock on the 1998 record in the materialized view, and computes the new value for the 1998 record based on the current value in the materialized view and Transaction #2 transition values. Transaction #2 obtains an X-lock on the 1998 record, updates the 1998 record, commits and releases its U- and X-locks. This the same essential execution pattern as Scenario #1.

(Cochrane, col. 8, lines 62 - col. 9, line 13) underline added.

This passage of Cochrane teaches two separate transactions to use the invention. One transaction performs reads, locks, computes and inserts on the MV table and a second transaction performs reads, locks, computes, and updates on the MV table. On the other hand, claim 5 recites "[a] statement to insert a new row or update an existing row in a database table, the statement implementing a process comprising the steps of: selecting from a source table a first column comprising a plurality of elements; selecting from a destination table a second column comprising a plurality of elements; determining a set of matching rows based upon the success of a comparison operation on an element in the first column and an element in the second column; determining a set of non-matching rows based upon the failure of a comparison operation on the first column element and the second column element; updating the destination table with the set of matching rows; and inserting into the destination table the set of non-matching rows, the statement comprising a single query language statement." As such, Cochrane does not disclose, teach, or suggest the limitations in claim 5.

Colby does not cure this deficiency. Colby teaches tagging, aggregating, and matching information in a pre-computed aggregate view of a materialized aggregate table using a first working data set to insert, a second working data set to remove, and a third working data set to update the materialized aggregate table. For example, Colby teaches:

> **FIG. 2** shows another method in accordance with the invention for incrementally maintaining a pre-computed aggregate view that is self-maintainable. The system receives the pre-computed aggregate view V that is self-maintainable, including its view definition V.sub.D and materialized aggregate table MV (step **202**). The system also receives changes to a base table of the pre-computed aggregate view, the changes being

represented as deletions and insertions (step **204**). The system tags insertions and deletions with distinguishing literals and combines them to produce .delta.F (deltaF) (step **206**). The system computes aggregations on .delta.F to produce aggregated change set .delta.G (deltaG) (step **208**). The system matches rows from .delta.G with rows in MV and produces .delta.J (deltaJ), which contains all rows from .delta.G with matched rows tagged with information from corresponding rows in MV and further contains non-matching rows from .delta.G tagged as non-matching (step **210**). (Matched rows are rows that have corresponding rows and non-matching rows are rows that do not have corresponding rows.) One way to tag rows in .delta.J as non-matching is to insert NULL values in MV columns. The system produces .delta.J' (deltaJ') by selecting from .delta.J rows identified as either: (i) matched rows or (ii) identified as non-matching but resulting from more base table changes that are insertions than deletions for the aggregated group represented by the row (step **212**). The system inserts new rows into MV, the new rows being constructed from rows (in .delta.J') identified as non-matching rows (step **214**). The system removes the identified rows from .delta.J' to produce .delta.I (deltaI) (also step **214**). The system deletes from MV rows matching rows in .delta.I, each row representing an aggregated group that has as many base table deletions as the sum of the number of base table insertions and the number of pre-modified base table rows for that group as stored in MV (step **216**). The system removes these identified rows from .delta.I to produce .delta.D (deltaD) (also step 216). The system updates rows in MV matching rows in .delta.D (step **218**).

(Colby, col. 7, line 62-col. 8, line 30) underline added.

This cites passage of Colby teaches aggregations, tagging, and matching of various data sets and using the various data sets to perform a *plurality of operations* (inserts, deletes and updates) *on the MV table*. However, Colby does not disclose, teach, or suggest "[a] statement to insert a new row or update an existing row in a database table, the statement implementing a process comprising the steps of: selecting from a source table a first column comprising a plurality of elements; selecting from a destination table a second column comprising a plurality of elements; determining a set of matching rows based upon the success of a comparison operation on an element in the first column and an element in the second column; determining a set of non-matching rows based upon the failure of a comparison operation on the first column element and the second column element; updating the destination table with the set of matching rows; and inserting into the destination table the set of non-matching rows, the statement comprising a single query language statement" as recited in claim 5. As neither Cochrane nor Colby, disclose, teach, or suggest the limitations in claim 5, Cochrane and Colby, can not be used be used to preclude patentability of claim 5 under 35 U.S.C. § 103.

5. Claim 21 recites sufficiently similar limitations to claim 5, and therefore is patentable over Cochrane and Colby for at least the same reasons.

6. Claims 6-8 and 22-24 depend on claims 5 and 21, and are therefore, patentable over Cochrane and Colby for at least the same reasons.

7. Claim 9 recites "[a] method for upserting a source table with a destination table, the method comprising: selecting from a source table a first column comprising a plurality of elements; selecting from a destination table a second column comprising a plurality of elements; updating a row in the destination table with a row from the source table upon the success of a comparison operation on an element in the first column of the row from the source table and an element in the second column of the row from the destination table; and inserting a row from the source table into the destination table upon the failure of a comparison operation on an element in the first column of the row from the source table and an element in the second column of the row from the destination table, the method using a single query language statement." Conversely Cochrane teaches a multiple transaction method. For example, Cochrane teaches:

> Consider Scenario #1a as a replacement for Scenario #1. Transaction #1 reads the materialized view, discovers that the 1998 record does not exist, and obtains an U-lock on either the next-key (e.g., the 1999 record) or EOF if the next-key does not exist. Transaction #1 then computes the value for the 1998 record based on Transaction #1 transition values. Transaction #1 obtains an X-lock on the next-key or EOF in the materialized view, inserts the new record into the materialized view, commits.and releases its U- and X-locks. Transaction #2 reads the materialized view, sees the 1998 record, obtains an U-lock on the 1998 record in the materialized view, and computes the new value for the 1998 record based on the current value in the materialized view and Transaction #2 transition values. Transaction #2 obtains an X-lock on the 1998 record, updates the 1998 record, commits and releases its U- and X-locks. This the same essential execution pattern as Scenario #1.

(Cochrane, col. 8, lines 62 - col. 9, line 13) underline added.

This passage of Cochrane teaches two separate transactions to perform the invention. On the other hand, claim 9 recites "[a] method for upserting a source table with a destination table, the method comprising: selecting from a source table a first column comprising a plurality of elements; selecting from a destination table a second column comprising a plurality of elements; updating a row in the destination table with a row from the source table upon the success of a comparison operation on an element in the first column of the row from the source table and an element in the second column of the row from the destination table; and inserting a row from the source table into the destination table upon the failure of a comparison operation on an element

in the first column of the row from the source table and an element in the second column of the row from the destination table, the method using a single query language statement." As two transaction cannot teach a single query language statement, Cochrane does not disclose, teach, or suggest the limitations in claim 9.

Colby does not cure this deficiency. Colby teaches tagging, aggregating, and matching information in a pre-computed aggregate view of a materialized aggregate table using a first working data sets to insert, a second working data set to remove, and a third working data set to update the materialized aggregate table. For example, Colby teaches:

> **FIG. 2** shows another method in accordance with the invention for incrementally maintaining a pre-computed aggregate view that is self-maintainable. The system receives the pre-computed aggregate view V that is self-maintainable, including its view definition V.sub.D and materialized aggregate table MV (step **202**). The system also receives changes to a base table of the pre-computed aggregate view, the changes being represented as deletions and insertions (step **204**). The system tags insertions and deletions with distinguishing literals and combines them to produce .delta.F (deltaF) (step **206**). The system computes aggregations on .delta.F to produce aggregated change set .delta.G (deltaG) (step **208**). The system matches rows from .delta.G with rows in MV and produces .delta.J (deltaJ), which contains all rows from .delta.G with matched rows tagged with information from corresponding rows in MV and further contains non-matching rows from .delta.G tagged as non-matching (step **210**). (Matched rows are rows that have corresponding rows and non-matching rows are rows that do not have corresponding rows.) One way to tag rows in .delta.J as non-matching is to insert NULL values in MV columns. The system produces .delta.J' (deltaJ') by selecting from .delta.J rows identified as either: (i) matched rows or (ii) identified as non-matching but resulting from more base table changes that are insertions than deletions for the aggregated group represented by the row (step **212**). The system inserts new rows into MV, the new rows being constructed from rows (in .delta.J') identified as non-matching rows (step **214**). The system removes the identified rows from .delta.J' to produce .delta.I (deltaI) (also step **214**). The system deletes from MV rows matching rows in .delta.I, each row representing an aggregated group that has as many base table deletions as the sum of the number of base table insertions and the number of pre-modified base table rows for that group as stored in MV (step **216**). The system removes these identified rows from .delta.I to produce .delta.D (deltaD) (also step 216). The system updates rows in MV matching rows in .delta.D (step **218**).

(Colby, col. 7, line 62-col. 8, line 30) underline added.

This cites passage of Colby teaches aggregation, tagging, and matching of various data sets and using the various data sets to perform *a plurality of operations* (inserts, deletes and updates) *on the MV table.* However, Colby does not disclose, teach, or suggest "[a] method for upserting a source table with a destination table, the method comprising: selecting from a source table a first column comprising a plurality of elements; selecting from a destination table a second column comprising a plurality of elements; updating a row in the destination table with a

row from the source table upon the success of a comparison operation on an element in the first column of the row from the source table and an element in the second column of the row from the destination table; and inserting a row from the source table into the destination table upon the failure of a comparison operation on an element in the first column of the row from the source table and an element in the second column of the row from the destination table, the method using a single query language statement" as recited in claim 9. As neither Cochrane nor Colby, disclose, teach, or suggest the limitations in claim 9, Cochrane and Colby, can not be used be used to preclude patentability of claim 9 under 35 U.S.C. § 103.

8. Claim 25 recites sufficiently similar limitations to claim 9, and therefore is patentable over Cochrane and Colby for at least the same reasons.

9. Claims 10-12 and 26-28 depend on claims 9 and 25, and are therefore, patentable over Cochrane and Colby for at least the same reasons.

10. Claim 13 recites "[a] computer implemented method for aggregating data in a database, comprising: parsing from a single command line, a command, a source table, a destination table, a source key, and a destination key; comparing the source key in each row of the source table with the destination key in each row of the destination table; determining a set of update rows based upon the success of a comparison operation performed on the source key and the destination key; determining a set of insert rows based upon the failure of a comparison operation performed on the source key and the destination key; updating the destination table with the set of update rows; and inserting into the destination table the set of insert rows." Conversely, Cochrane teaches a multiple transaction method. For example, Cochrane teaches:

> Consider Scenario #1a as a replacement for Scenario #1. Transaction #1 reads the materialized view, discovers that the 1998 record does not exist, and obtains an U-lock on either the next-key (e.g., the 1999 record) or EOF if the next-key does not exist. Transaction #1 then computes the value for the 1998 record based on Transaction #1 transition values. Transaction #1 obtains an X-lock on the next-key or EOF in the materialized view, inserts the new record into the materialized view, commits.and releases its U- and X-locks. Transaction #2 reads the materialized view, sees the 1998 record, obtains an U-lock on the 1998 record in the materialized view, and computes the new value for the 1998 record based on the current value in the materialized view and Transaction #2 transition values. Transaction #2 obtains an X-lock on the 1998 record,

updates the 1998 record, commits and releases its U- and X-locks. This the same essential execution pattern as Scenario #1.

(Cochrane, col. 8, lines 62 - col. 9, line 13) underline added.

This passage of Cochrane teaches two separate transactions to perform the invention. On the other hand, claim 13 recites "[a] computer implemented method for aggregating data in a database, comprising: parsing from a single command line, a command, a source table, a destination table, a source key, and a destination key; comparing the source key in each row of the source table with the destination key in each row of the destination table; determining a set of update rows based upon the success of a comparison operation performed on the source key and the destination key; determining a set of insert rows based upon the failure of a comparison operation performed on the source key and the destination key; updating the destination table with the set of update rows; and inserting into the destination table the set of insert rows." As such Cochrane does not disclose, teach, or suggest the limitations in claim 13.

Colby does not cure this deficiency. Colby teaches tagging, aggregating, and matching information in a pre-computed aggregate view of a materialized aggregate table using a first working data set to insert, a second working data set to remove, and a third working data set to update the materialized aggregate table. For example, Colby teaches:

FIG. 2 shows another method in accordance with the invention for incrementally maintaining a pre-computed aggregate view that is self-maintainable. The system receives the pre-computed aggregate view V that is self-maintainable, including its view definition V.sub.D and materialized aggregate table MV (step **202**). The system also receives changes to a base table of the pre-computed aggregate view, the changes being represented as deletions and insertions (step **204**). The system tags insertions and deletions with distinguishing literals and combines them to produce .delta.F (deltaF) (step **206**). The system computes aggregations on .delta.F to produce aggregated change set .delta.G (deltaG) (step **208**). The system matches rows from .delta.G with rows in MV and produces .delta.J (deltaJ), which contains all rows from .delta.G with matched rows tagged with information from corresponding rows in MV and further contains non-matching rows from .delta.G tagged as non-matching (step **210**). (Matched rows are rows that have corresponding rows and non-matching rows are rows that do not have corresponding rows.) One way to tag rows in .delta.J as non-matching is to insert NULL values in MV columns. The system produces .delta.J' (deltaJ') by selecting from .delta.J rows identified as either: (i) matched rows or (ii) identified as non-matching but resulting from more base table changes that are insertions than deletions for the aggregated group represented by the row (step **212**). The system inserts new rows into MV, the new rows being constructed from rows (in .delta.J') identified as non-matching rows (step **214**). The system removes the identified rows from .delta.J' to produce .delta.I (deltaI) (also step **214**). The system deletes from MV rows matching rows in .delta.I, each row representing an aggregated group that has as many base table deletions as the sum of the number of base table insertions and the number of pre-modified base table rows for that group as

> stored in MV (step **216**). The system removes these identified rows from .delta.I to produce .delta.D (deltaD) (also step 216). The system updates rows in MV matching rows in .delta.D (step **218**).

(Colby, col. 7, line 62-col. 8, line 30) underline added.

This cites passage of Colby teaches aggregations, tagging, and matching of various data sets and using the various data sets to perform a *plurality of operations* (inserts, deletes and updates) *on the MV table*. However, Colby does not disclose, teach, or suggest "[a] computer implemented method for aggregating data in a database, comprising: parsing from a single command line, a command, a source table, a destination table, a source key, and a destination key; comparing the source key in each row of the source table with the destination key in each row of the destination table; determining a set of update rows based upon the success of a comparison operation performed on the source key and the destination key; determining a set of insert rows based upon the failure of a comparison operation performed on the source key and the destination key; updating the destination table with the set of update rows; and inserting into the destination table the set of insert rows" as recited in claim 13. As neither Cochrane nor Colby, either alone or in combination, disclose, teach, or suggest the limitations in claim 13, Cochrane, and Colby, neither alone nor in combination can be used be used to preclude patentability of claim 13 under 35 U.S.C. § 103.

11. Claim 29 recites sufficiently similar limitations to claim 13, and therefore is patentable over Cochrane and Colby for at least the same reasons.

12. Claims 14-16 and 30-32 depend on claims 13 and 29, and are therefore, patentable over Cochrane and Colby for at least the same reasons.

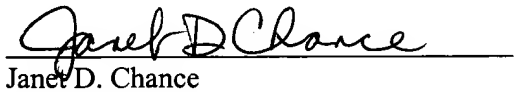
CONCLUSION

On the basis of the above remarks, reconsideration and allowance of the claims is believed to be warranted and such action is respectfully requested. If the Examiner has any questions or comments, the Examiner is respectfully requested to contact the undersigned at the number listed below.

The Commissioner is hereby authorized to charge any additional fees, or credit any overpayments, to Bingham McCutchen Deposit Account No. 50-2518, billing reference OI7011122001.

Respectfully submitted,
Bingham McCutchen LLP

Dated: January 18, 2005

By: 
Janet D. Chance
Reg. No. 55,048

Three Embarcadero Center, Suite 1800
San Francisco, CA 94111-4067
Telephone: (650) 849-4904
Telefax: (650) 849-4800